isitech
CAMPUS DU NUMÉRIQUE

Atelier professionnel

Mise en place d'un EtherChannel supervisé avec alertes automatisées

Avec liaison multi-lien (EtherChannel / Port Channel)

Auteur : Konate Mamady

Contexte : Projet AP3 – BTS SIO SISR

debian

Grafana

SOMMAIRE

1. Objectifs du projet

L’objectif de ce projet est d’améliorer la performance réseau entre un switch d'accès et un switch principal, tout en mettant en place une supervision proactive à l’aide d’outils modernes.

Objectifs techniques :

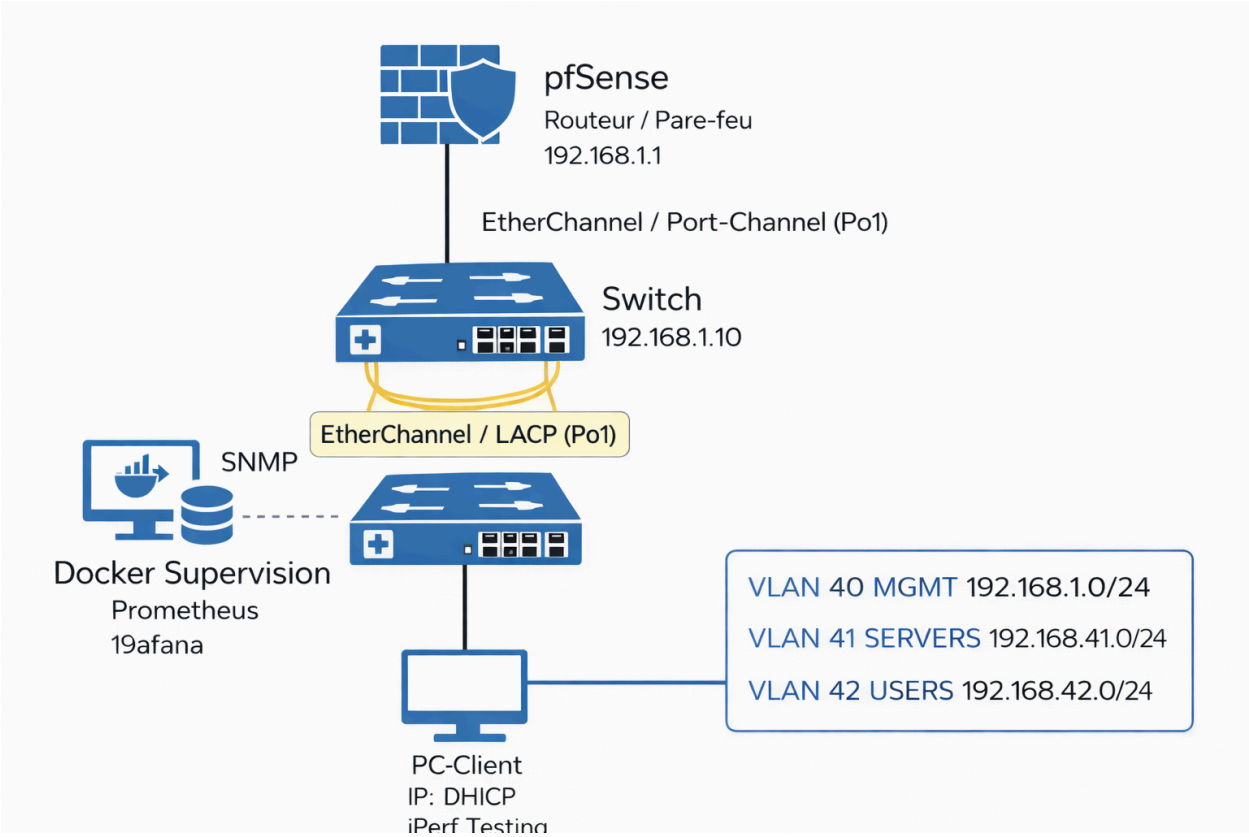
- Agrégation de liens via EtherChannel
- Supervision en temps réel du trafic réseau
- Détection automatique des surcharges

- Envoi d’alertes par **e-mail**
- Tests de montée en charge avec **iPerf3**

*iPerf3 : outil en ligne de commande utilisé pour mesurer les performances d’un réseau

*EtherChannel : ’agréger plusieurs liens physiques en un seul lien logique (pour assurer la redondance)

2. 🗺️ Schéma réseau



3. 🧱 Tableau d'adressage

Équipement	Rôle	Interface	VLAN	Adresse IP	Masque
pfSense	Routeur/Firewall	LAN	40	192.168.1.1	/24
Switch d'accès	Accès utilisateurs	VLAN Interface	40	192.168.1.10	/24
SRV-SUPERVISION debian	Supervision Docker	v	40	192.168.1.47	/24
PC-Client	Test iPerf	ETH	42	DHCP	/24

---

4. 🧰 Moyens matériels et logiciels

Matériel :

- Switch Cisco
- 1x Serveur Debian 12
- 1x PC client (iperf3)
- 1x Routeur pfSense
-

Logiciels & Protocoles :

- **Docker**, **Docker Compose**
- **Prometheus**, **Grafana**, **SNMP Exporter**
- **iPerf3**
- **SNMP v2c**

---

5. Procédure

5.1 Configuration des Switches

Configuration EtherChannel (SW d'accès et SW principal) :

```
SW9#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
SW9(config)#interface range GigabitEthernet1/0/13 - 14
SW9(config-if-range)#description Lien LACP vers switch
SW9(config-if-range)#switchport mode trunk
SW9(config-if-range)#switchport trunk allowed vlan 40-49
SW9(config-if-range)#chanel-group 1 mode active
                         ^
% Invalid input detected at '^' marker.

SW9(config-if-range)#channel-group 1 mode active
Creating a port-channel interface Port-channel 1

SW9(config-if-range)#
```

```
SW9(config-if-range)#channel-group 1 mode active
Creating a port-channel interface Port-channel 1

SW9(config-if-range)#exit
SW9(config)#interface Port-channel1
SW9(config-if)#description lien EtherChannel vers le coeur de reseau
SW9(config-if)#switchport mode trunk
SW9(config-if)#switchport trunk allowed vlan 40-49
SW9(config-if)#exit
SW9(config)#
```

🧷 **VLANs utilisés :**

```
vlan 40
 name MGMT
vlan 41
 name SERVERS
vlan 42
 name USERS
```

```
SW9(config-if-range)#exit
SW9(config)#interface Port-channel1
SW9(config-if)#description lien EtherChannel vers le coeur de reseau
SW9(config-if)#switchport mode trunk
SW9(config-if)#switchport trunk allowed vlan 40-49
SW9(config-if)#exit
SW9(config)#
SW9(config)#vlan 40
SW9(config-vlan)#exit
SW9(config)#vlan 41
SW9(config-vlan)#name Serveur
SW9(config-vlan)#vlan 42
SW9(config-vlan)#name Client
SW9(config-vlan)#exit
SW9(config)#
```

🩹 **Activation SNMP :**

```
SW9#
SW9#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
SW9(config)#snmp-server community public RO
SW9(config)#acces-list 10 permit 192.168.1.47
                ^
% Invalid input detected at '^' marker.

SW9(config)#access-list 10 permit 192.168.1.47
SW9(config)#snmp-server community public RO 10
SW9(config)#
```

Création d'une ACL nommée 10 autorisant l'adresse IP `192.168.1.47` (mon serveur de supervision).

Sécuriser l'accès SNMP en restreignant aux IP autorisées.

5.2 Mise en place de la stack de supervision

Installation de Docker :

“apt install docker-ce docker-compose-plugin -y”

- **Fichiers dans `/root/supervision/` :**
  - `prometheus.yml`
  - `docker-compose.yml`

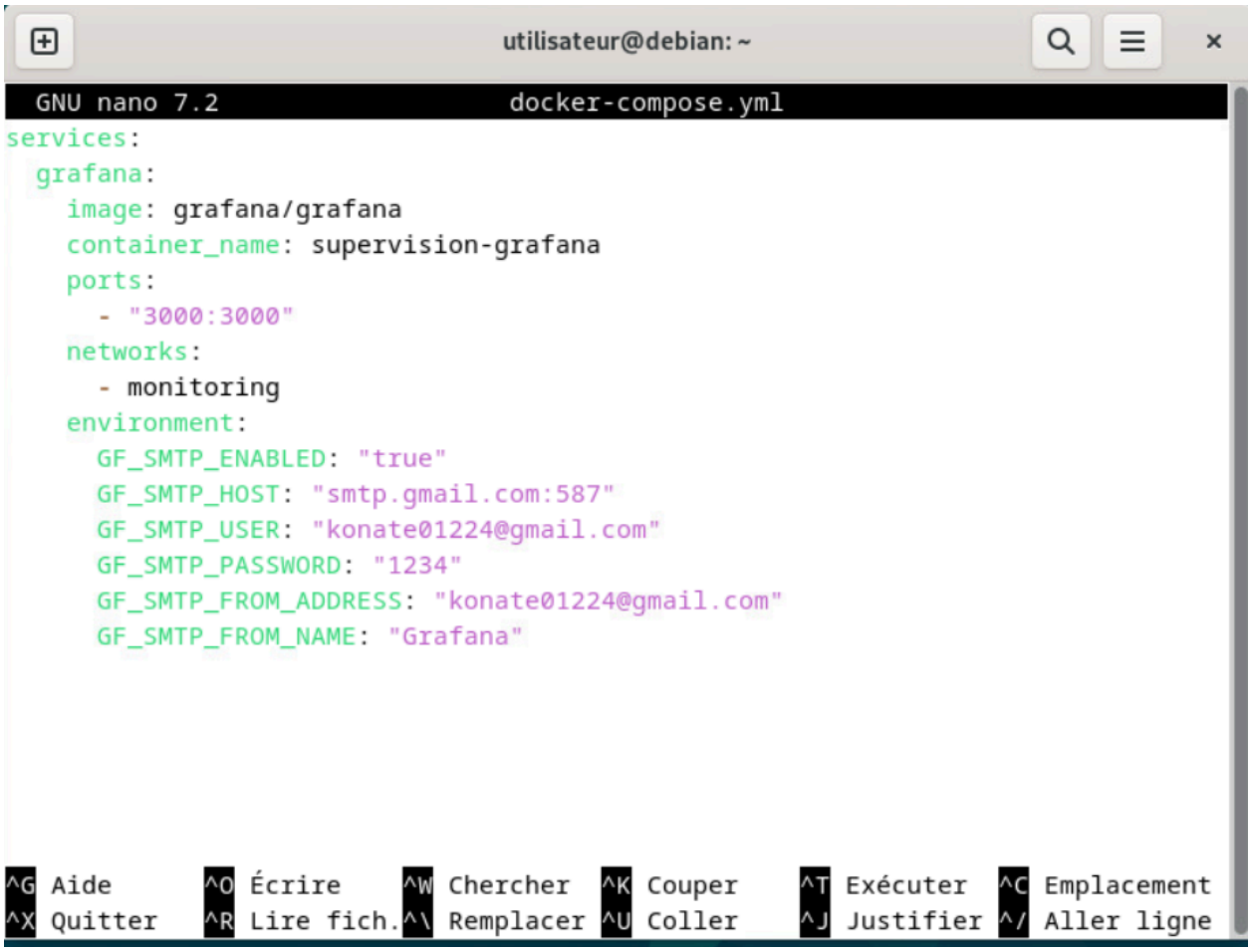
📁 **prometheus.yml** :

```
  GNU nano 7.2                        prometheus.yml
global:
  scrape_interval: 10s

scrape_configs:
  - job_name: 'snmp-switch'
    static_configs:
      - targets: ['192.168.1.10']
    metrics_path: /snmp
    params:
      module: [if_mib]
    relabel_configs:
      - source_labels: [__address__]
        target_label: __param_target
      - source_labels: [__param_target]
        target_label: instance
      - target_label: __address__
        replacement: snmp-exporter:9116


                          [ Lecture de 17 lignes ]
^G Aide       ^O Écrire     ^W Chercher   ^K Couper     ^T Exécuter   ^C Emplacement
^X Quitter    ^R Lire fich. ^\ Remplacer  ^U Coller     ^J Justifier  ^/ Aller ligne
```

📁 **`docker-compose.yml`** (extrait `grafana` avec SMTP) :



5.3 Configuration de Grafana

1. Accès : http://192.168.1.47:3000
(correspond à l'IP du serveur de supervision)

2. Login : admin / admin

3.

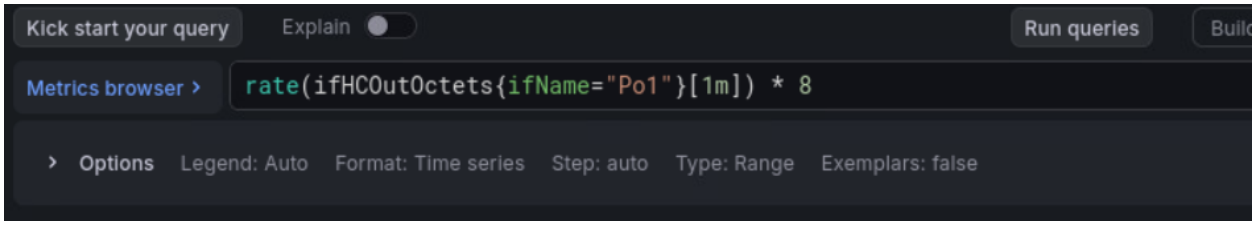
🔹 **Création du dashboard :**

Query IN :

```
rate(ifHCInOctets{ifName="Po1"}[1m]) * 8
```

Query OUT :

```
rate(ifHCOutOctets{ifName="Po1"}[1m]) * 8
```



5.4 🚨 Configuration des alertes

1. Alertes créées :
    - Trafic entrant > 80 Mb/s
    - Trafic sortant > 80 Mb/s
2. Durée : 1 minute
3. Canal d'alerte : e-mail via SMTP Gmail

4. 

5.5 🧪 Test de validation avec iPerf3

Serveur (Debian) :

```
iperf3 -s
```

Client (poste de test) :

```
iperf3 -c 192.168.1.7 -u -b 200M -t 300
```

6. ✅ Conclusion

Ce projet m’a permis de :

- Mettre en œuvre une agrégation de liens (EtherChannel) stable et performante
- Superviser les métriques réseau en temps réel
- Déclencher des alertes de surcharge automatiquement
- Utiliser des outils modernes (Docker, Grafana, Prometheus)